
Conformative Filtering for Implicit Feedback Data

Farhan Khawar*, Nevin L. Zhang*, Jinxing Yu
Hong Kong University of Science and Technology
{fkhawar,lzhang,jyuat}@cse.ust.hk

Abstract

Implicit feedback is the simplest form of user feedback that can be used for item recommendation. It is easy to collect and domain independent. However, there is a lack of negative examples. Existing works circumvent this problem by making various assumptions regarding the unconsumed items, which fail to hold when the user did not consume an item because she was unaware of it. In this paper we propose Conformative Filtering (CoF) as a novel method for addressing the lack of negative examples in implicit feedback. The motivation is that if there is a large group of users who share the same taste and none of them consumed an item, then it is highly likely that the item is irrelevant to this taste. We use Hierarchical Latent Tree Analysis (HLTA) to identify taste-based user groups, and make recommendations for a user based on her memberships in the groups. Experiments on real-world datasets from different domains show that CoF has superior performance compared to other baselines and more than 10% improvement in Recall@5 and Recall@10 is observed.

1 Introduction

With the advent of the online marketplace, an average user is presented with an un-ending choice of items to consume. These items can be the books bought, web-pages clicked, songs listened, or the movies watched etc. Online stores and content providers no longer have to worry about the limited shelf space to display their items. However, too much choice is not always a luxury. It can often be an unwanted distraction or it can make it difficult

for a user to find the items she desires. Therefore, from the space of all items we need to filter relevant items for a user.

Collaborative filtering (CF) [Goldberg *et al.*, 1992] is one commonly used technique to deal with the problem. Most research work on CF focuses on explicit feedback data, where users provide explicit ratings for items [Koren and Bell, 2015]. Items with high ratings are preferred over those with low ratings. In other words, items with high ratings are positive examples, while those with low ratings are negative examples. Unrated items are treated as missing data.

In practice, one often encounters implicit feedback data, where users do not explicitly rate items [Nichols, 1997]. Recommendations need to be made based on user’s activities such as clicks, page views, and purchase actions. Those are positive-only data and contain information regarding which items were consumed. There is no information about the unconsumed items. In other words, there are no negative examples.

In comparison with CF with explicit feedback, CF with implicit feedback has received considerably less attention. A key issue with implicit feedback is how to deal with the lack of negative examples [Ricci *et al.*, 2015]. We are unsure whether the user didn’t consume an item because she didn’t like it or because she never saw it. In this paper, we propose a novel method for addressing this issue. We start by identifying user groups with the same tastes. By a *taste* we mean the tendency to consume a certain collection of items such as comedy movies, pop songs, or spicy food. Those taste-based groups give us a nice way to deal with the lack of negative examples. While it is not justifiable to assume that non-consumption means disinterest for an individual user, it is relatively more reasonable to make that assumption for a group of users with the same taste: If many users share a taste and none of them have consumed an item before, then it is likely that the group is

Corresponding authors

not interested in the item.

We use HLTA [Chen *et al.*, 2016; Liu *et al.*, 2014] to identify taste-based user groups. When applied to implicit feedback data, HLTA obtains a hierarchy of binary latent variables by: (1) Identifying item co-consumption patterns (groups of items that tend to be consumed by the same customers, not necessarily at the same time) and introducing a latent variable for each pattern; (2) Identifying co-occurrence patterns of those patterns and introducing a latent variable; (3) Repeating step 2 until termination. Each of the latent variables identifies a soft group of users, just as the concept “intelligence” denotes a class of people.

To make recommendations, we choose the user groups from a certain level of the hierarchy and characterize each group by aggregating recent behaviors of its members. For a particular user, we perform inference on the learned model to determine her memberships in the groups, and predict her preferences by combining her memberships in the groups and the group characteristics. We call this method *Conformative Filtering* because a user is expected to conform to the behaviors of the groups she belongs to.

The rest of the paper is organized as follows: We start by reviewing existing methods for dealing with lack of negative feedback in Section 2 and the basics of latent tree models in Section 3. In Section 4 we show how latent tree models can be used to identify the taste-based user groups, and in Section 5 we present our conformative filtering method. Empirical results are given in Section 6, followed by concluding remarks.

2 Related Work

CoF is similar to user-kNN [Ricci *et al.*, 2010] in that they both predict a user’s preferences based on past behaviors of similar users. There are two important differences. First, when a user belongs to multiple taste groups, as is usually the case, CoF uses information from all the users in those groups, while user-kNN uses information only from the users who are in all groups. To put it another way, CoF uses the union of the groups, while user-kNN uses their intersection. This is illustrated in Figure 1. Second, user-kNN is not model-based whereas CoF is. More specifically, the taste groups are the latent factors. An item is characterized by the frequencies it was consumed by members of the groups, and a user is characterized by her memberships in the groups. In comparison with matrix factorization, the latent factors in CoF are more interpretable. They also offer more flexibilities. For example, we can use recent behaviors of group members, instead of their entire consumption his-

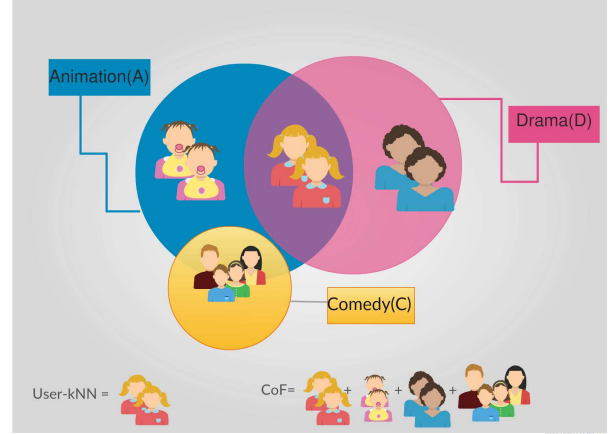


Figure 1: Groups of people used by user-kNN and CoF to recommend items for a user who has three tastes in videos: Animation, Comedy and Drama. Each circle in the Venn diagram represents a group of people who have a taste for the video genre indicated. User-kNN uses the intersection of the three groups while CoF uses the union of them.

stories, when predict future behavior of the group.

Several model-based methods have been previously proposed to deal with the lack of negative examples in implicit feedback. One idea is to have two parts in the loss function for matrix factorization. Let us call a user-item pair (u, i) a *consumption pair* if user u has consumed item i before, and a *non-consumption pair* otherwise. The first part of the loss function encourages the predicted score \hat{r}_{ui} to be close to 1 for consumption pairs, while the second part encourages the predicted score \hat{r}_{ui} to be close to 0 for non-consumption pairs. The second part is associated with lower weights than the first part. Thus, non-consumption is essentially viewed as disinterest with low confidence [Hu *et al.*, 2008; Pan *et al.*, 2008].

Alternatively, one can assign the same weight to the two parts of the loss function but include in the second part a subset of non-consumption pairs that are randomly sampled [Pan *et al.*, 2008]. Heuristics are used to maximize the chance of the sampled non-consumption pairs being truly disinterest pairs. For example, one can assign higher sampling probabilities to non-consumption pairs (u, i) where the item i is unpopular or the user u is active.

Another idea is to assume that a user prefers consumed items over unconsumed items, as done in Bayesian personalized ranking (BPR) [Rendle *et al.*, 2009]. The idea is realized by employing a loss function that depend negatively on the difference $\hat{r}_{ui} - \hat{r}_{uj}$ between predicted scores for consumption pairs (u, i) and non-consumption pairs (u, j) for the same user u . Consequently, minimiz-

ing the loss function leads to high scores for consumption pairs and low scores for non-consumption pairs.

The assumptions behind all these methods are about the preferences of individual users. They fail to hold when a user u would have liked an item i but did not consume it only because she was unaware of it. In that case, it is incorrect to assume user u is not interested in item i , even with low confidence; it would be a mistake if the pair (u, i) is sampled as a negative example; and it is wrong to assume u prefers all her consumed items to item i . In contrast, the assumption behind CoF is about the preferences of groups of users. If a group is large enough, it is relatively safe to assume that most of the items have come to the attention of at least one of the group members, and hence relatively reasonable to assume that the items not consumed by any group members are not of interest to the group.

3 Basics of Latent Tree Models

In this section, we briefly review the basics of latent tree models. The materials are borrowed from [Zhang and Poon, 2017].

A *latent tree model (LTM)* is a tree-structured Bayesian network [Pearl, 1988], where the leaf nodes represent observed variables and the internal nodes represent latent variables. An example is shown in Figure 1 (a). All variables are assumed to be binary in this paper. The model parameters include a marginal distribution for the root Y_1 and a conditional distribution for each of the other nodes given its parent. The product of the distributions defines a joint distribution over all the variables.

By changing the root from Y_1 to Y_2 in Figure 1 (a), we get another model shown in (b). The two models are *equivalent* in the sense that they represent the same set of distributions over the observed variables X_1, \dots, X_5 [Zhang, 2004]. It is not possible to distinguish between equivalent models based on data. This implies that edge orientations in LTMs are unidentifiable. It therefore makes more sense to talk about undirected LTMs, which is what we do in this paper. One example is shown in Figure 1 (c). It represents an equivalent class of directed models, which includes the two models shown in (a) and (b) as members. In implementation, an undirected model is represented using an arbitrary directed model in the equivalence class it represents.

To learn an LTM from a dataset, one needs to determine: (1) the number of latent variables, (2) the connections among all the variables, and (3) the probability distributions. A host of algorithms have been proposed. In particular, [Chen *et al.*, 2016; Liu *et al.*, 2014] have developed an algorithm for learning *hierarchical latent tree*

Table 1: Information about the latent variables Z_{13} , that represents the taste for three action-adventure-thriller movies, and Z_{1147} , that represents the taste for three children-animation movies. The marginal distributions $P(Z_{13})$ and $P(Z_{1147})$ are shown in parentheses. The rest of the rows show the probabilities of the movies having been watched by users with and without the taste respectively.

	$Z_{13} = s1$	$Z_{13} = s0$
Action-Adventure-Thriller	(0.21)	(0.79)
<i>Armageddon</i>	0.610	0.055
<i>GoldenEye</i>	0.588	0.013
<i>ConAir</i>	0.635	0.014

	$Z_{1147} = s1$	$Z_{1147} = s0$
Children-Comedy	(0.09)	(0.91)
<i>GreatMuppetCaperThe</i>	0.456	0.009
<i>PetesDragon</i>	0.450	0.004
<i>MuppetsTakeManhattanThe</i>	0.457	0.005

models (HLTMs), in the context to topic detection, that has a layer of observed variables at the bottom and multiple layers of latent variables on the top. Their algorithm is called HLTA ¹.

4 Taste Group Detection

HLTA can be used to analyze implicit feedback data if we regard the items as words and the consumption history of a user as a document. We have run it on the MovieLens1M dataset. A part of the resulting model is shown in Figure 3. The variables at the leaves are binary variables that indicate whether movies are consumed. The other variables are latent variables. Each latent variable partitions the users into two clusters. Hence, multiple partitions of the users are obtained [Chen *et al.*, 2012; Liu *et al.*, 2015]. Users in the same clusters have the tendency to consume same subsets of the items. So, multiple tastes are detected, with one identified by each latent variable.

For example, the movies “Armageddon”, “Golden Eye” and “Con Air” are grouped under the latent variable Z_{13} , which reveals the pattern that the three movies tend to be co-consumed, i.e., watched by the same viewers. Z_{13} partitions all the users into two soft clusters. The relevant numerical information is shown in Table 1. The first cluster $Z_{13} = s1$ consists of 21% of the users. Users in this cluster have high probabilities of watching these movies. So, they have a taste for the three movies, which is a sub-collection of action-adventure-thriller movies. In contrast, users in the second cluster $Z_{13} = s0$ have low

¹An implementation of HLTA is available at <https://github.com/kmpoon/hlta>.

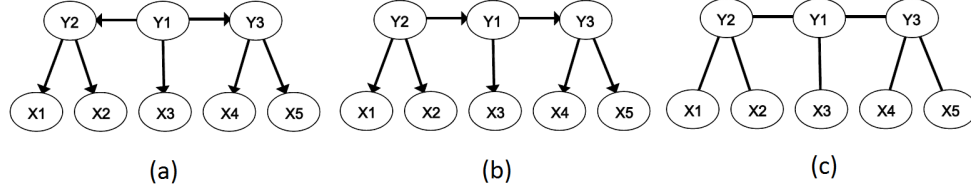


Figure 2: The undirected latent tree model in (c) represents an equivalent class of directed latent tree models, which includes (a) and (b) as members.

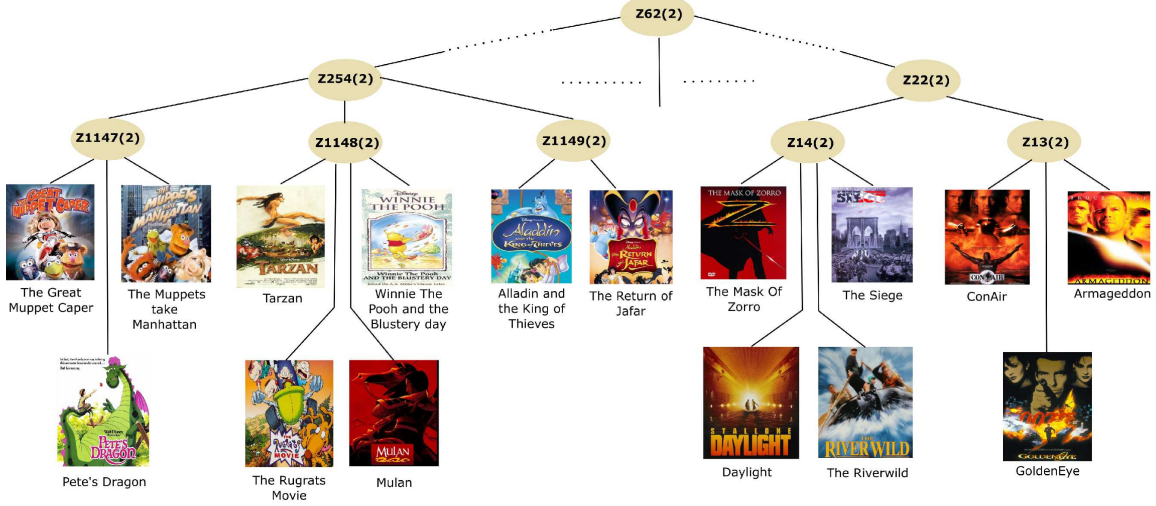


Figure 3: Parts of the hierarchical latent tree model obtained from Movielens1M dataset. The level-1 latent variables reveal co-consumption of items by users and identify user tastes for various subsets of items. Latent variables at higher levels reveal co-occurrence of the tastes at the level below and identify more broad tastes.

probabilities of watching these movies and hence they do not possess the taste.

Similarly, Z_{14} identifies a group of users with a taste for the movies “The Seige”, “Mask of Zorro”, “Daylight” and “The River Wild”. It is grouped together with Z_{13} under Z_{22} , which indicates that the two tastes tend to be co-possessed, and Z_{22} identifies the users who tend to have both tastes.

Different parts of the model represent different genres and a variety of tastes are detected. For example, we see that Z_{1148} represents users who consume children-animation movies like “Tarzan”, “Rugrats”, “Mulan” and “Winnie the Pooh and the Blustery Day” and Z_{1147} represents users with taste for the children-comedy movies “The Great Muppet Caper”, “Pete’s Dragon” and “The Muppets take Manhattan”. Similarly, other parts of the model represent users who have taste for different genres like horror, documentary, sci-fi, noir etc.

5 Conformative filtering

Suppose we have learned an HLTM m from implicit feedback data and suppose there are K latent variables

on the l -th level of the model, each with two states s_0 and s_1 . Denote the latent variables as Z_{l1}, \dots, Z_{lK} . They give us K taste-based user groups $Z_{l1} = s_1, \dots, Z_{lK} = s_1$, which will sometimes be denoted as G_1, \dots, G_K for simplicity. In this section, we show how these taste groups can be used for item recommendation.

5.1 Taste Group Characterization

A natural way to characterize a user group is to aggregate past behaviors of the group members. The issue is somewhat complex for us because our user groups are soft clusters. Let $\mathbb{I}(i|u, \mathcal{D})$ be the indicator function which takes value 1 if user u has consumed item i before, according to the dataset \mathcal{D} , and 0 otherwise. We determine the preference of a taste group G_k (i.e., $Z_{lk} = s_1$) on an item i as follows:

$$p(i|G_k, \mathcal{D}) = \frac{\sum_u \mathbb{I}(i|u, \mathcal{D}) P(Z_{lk} = s_1|u, m)}{\sum_u P(Z_{lk} = s_1|u, m)}, \quad (1)$$

where $P(Z_{lk} = s_1|u, m)$ is the probability of user u being in the soft cluster $Z_{lk} = s_1$ according to the model m , and the summations are over all users.

Note that $p(i|G_k, \mathcal{D}) = 0$ if no users in G_k have consumed the item i before. In other words, we assume that a group is not interested in an item if none of the group members have consumed the item before. There is a risk of treating an item as a negative example because none of the group members have seen it. We argue that the probability of this happening is low when the group is large.

Theorem 1. *Let there be m people in a group and N items in total. Suppose each person picks n items with replacement randomly and independently. Let X be the total number of different items picked. For any two numbers $q, p \in [0, 1]$, we have*

$$P(X \geq qN) \geq p,$$

$$\text{if } m \geq \frac{\log\left(1 - q - \sqrt{\frac{-\log(1-p)}{2N}}\right)}{n \log(1 - 1/N)}.$$

For example, if each user randomly picks 20 items from a total of 1000, then with a group of *only* 136 users we can ensure that the probability of at least 90% of the items being seen by at least one person is at least 0.9.

Also note that the reason we determine the preferences of a group G_k is that we want to predict future behavior of the group members. As such, we might want to use recent behaviors of the group members instead of their entire consumption histories. For example, we can choose to use the latest H consumptions for each user. The resulting subset of data is denoted as \mathcal{D}_H . As will be seen later, H has an impact on the quality of item recommendations.

5.2 User Characterization

A user u is characterized using her memberships in the K clusters, i.e.,

$$\mathbf{u} = (P(Z_{l1} = s1|u, m), \dots, P(Z_{lK} = s1|u, m)). \quad (2)$$

Note that m is a tree-structure model. All the K posterior probability values can be calculated by propagating messages over the tree twice [Pearl, 1988]. It takes time linear in the number of variables in the model, and hence linear in the number of items.

5.3 Item Recommendation

To make recommendations, we first compute a score for each user-item pair, and, for each user, we recommend the items with the highest scores. The score \hat{r}_{ui} for a user-item pair (u, i) is computed by combining the

Table 2: Statistics about the datasets

Ta-feng	Users	Items	Sparsity
train	27574	22226	99.907%
validation	12261	15206	99.934%
test	13191	14561	99.936%
Amazon Baby			
train	4078	2367	99.732%
validation	1635	1817	99.813%
test	1544	1788	99.799%
Movielens1M			
train	4870	3633	96.043 %
validation	1352	3426	96.761%
test	1512	3482	97.150%

taste group characterizations and the memberships of u in those groups:

$$\hat{r}_{ui} = \sum_{k=1}^K p(i|G_k, \mathcal{D}_H) P(Z_{lk} = s1|u, m). \quad (3)$$

The score is the inner product of two vectors — the user characterization vector \mathbf{u} and the item characterization vector $\mathbf{i} = (p(i|G_1, \mathcal{D}_H), \dots, p(i|G_K, \mathcal{D}_H))$; that characterizes item i in terms of the preference scores of the K taste groups for i .

6 Experiments

We performed experiments on three publicly available datasets from different domains. Each dataset comprised only of (*user*, *item*, *time-stamp*) tuples. Following [Wu *et al.*, 2017], we split a dataset into training, validation and test subsets by time. This is so that all the training instances came before all the testing instances, which matches real-world scenarios better than splits that do not consider time. We tested on several splits of the datasets and the results were similar. In the following, we will only report the results on the split with 70% of the data for training, 15% for validation and 15% for test.

6.1 Datasets

Here are the three datasets used in our experiments:

- **Movielens1M**² is a dataset that contains the ratings given by users to the movies they watched.
- **Amazon baby** [McAuley *et al.*, 2015] is a dataset that consists of users providing ratings for the baby products they bought from Amazon. Following the suggestion of the dataset providers, we retained only users and items with more than 5 ratings.

²<https://grouplens.org/datasets/movielens/1m/>

- **Ta-feng**³ is an implicit feedback supermarket dataset which consists of buy events. Where a user buying an item from the supermarket is an event.

The first two are explicit feedback data and were converted to implicit feedback data by ignoring the explicit ratings. Then, all the events of a user rating an item indicate that a user consumed (watched, bought) this item. Statistics about the datasets are found in Table 2.

6.2 Setup

We compared CoF against the four popular baselines: weighted-user-KNN (W-UkNN), weighted-item-KNN (W-IkNN), weighted regularized matrix factorization (WRMF) [Hu *et al.*, 2008], and Bayesian personalized ranking matrix factorization (BPRMF) [Rendle *et al.*, 2009]. As mentioned in Section 2, WRMF and BPRMF were specifically proposed to deal with the lack of negative examples in implicit feedback. MyMediaLite [Gantner *et al.*, 2011] implementation was used for all baselines.

Parameters of all methods were tuned based on the validation set. After parameter tuning we re-trained all the models on the train and validation set and tested on the test set. For WRMF and BPRMF, we performed grid search for the latent factors over $F \in \{10, 20, 40, 80, 160\}$, and for regularization over $\lambda = \{10^{-4}, 10^{-3}, \dots, 10^2\}$. For W-UkNN and W-IkNN, k was searched from the set $\{10, 20, 40, 80, 160, 320, 500\}$. For CoF we searched for H over $\{2, 3, 5, 10, 20, \dots, 100\}$ and l over the number of levels in m . The number of latent factors on each level are automatically determined during model construction.

6.3 Evaluation Measures

We used two popular metrics to evaluate the quality of the recommended lists for implicit feedback. They are briefly outlined below:

- **Recall@ R** : It is the fraction, among all items consumed by a user, of those that are placed at the top R positions in a recommended list.
- **NDCG**: Discounted cumulative gain is defined as $DCG = \sum_{i=1}^P \frac{rel_i}{\log_2(i+1)}$, where P is the total number of recommended items and $rel_i \in \{0, 1\}$ indicates whether the item at position i is consumed by the user. NDCG is DCG normalized by the ideal DCG of the list.

While Recall@ R was calculated on the top R items, NDCG was calculated over the entire recommendation list that involved all unconsumed items.

Note that Precision is known to be unsuitable for implicit data [Wang and Blei, 2011], because it treats all unconsumed items as negative examples. Therefore, Precision based metrics (Precision@ R and Mean Average Precision) are not used in this paper.

6.4 Main Results

The result are shown in Table 3. Recall@ R is arguably the most important metric for implicit feedback. In the real-world scenario, one can recommend only a few items to a user, and one would hope the list contains as many items of interest to the user as possible. This is exactly what Recall@ R measures.

In terms of Recall@ R , CoF performed drastically better than all the baselines. For example, the Recall@5 and Recall@10 scores of CoF are more than 10% higher than those of the baselines on all the datasets. Note that the second best method is user-kNN, which is not a model-based method. If we compare CoF to only model-based alternatives, the improvement is more than 20%.

Moreover, for Movilens1M, CoF outperforms all competing methods by a large margin. This shows that when the data is less sparse, CoF is able to extract meaningful information much more effectively than other methods.

In terms of NDCG, CoF also performed better than the baselines in all cases. NDCG gives high score to recommendation lists in which relevant items have higher rank in the list. This indicates that the recommend lists (over all items) produced by CoF are of higher quality than those given by the baselines.

User-kNN comes the second in most cases and beats CoF in one case. It should be noted that CoF is model-based. The user and item characterization vectors can be computed offline. Hence, it scales up better than user-kNN.

6.5 Impact of Parameters

There are two parameters in CoF: l and H . The first parameter l determines which level of the hierarchical model produced by HLTA to use. The larger the l , the fewer the number of latent factors. The second parameter H denotes the number of latest consumptions of a user to be used when characterizing user groups. Although both parameters are selected using the validation, it would be interesting to gain some insight of their impact on performance.

³<http://www.bigdatalab.ac.cn/benchmark/bm/dataset/> Figure 4(a) shows the Recall@5 scores on Ta-feng as a

Table 3: Performance evaluation on test-set: **boldface** denotes the best performance, underline denotes the second best performance and parentheses contain the percentage improvement by CoF over the second best result. CoF outperforms other methods in all but one test case. Substantial improvement in Recall@5 and Recall@10 is observed.

Ta-feng	Recall@5	Recall@10	Recall@20	NDCG
CoF($l=1, H=40, K=2004$)	0.05582 _(13.4%)	0.06627 _(12.8%)	0.07661 _(6.1%)	0.23359 _(17.4%)
WRMF($F=10, \lambda=10$)	0.01854	0.02970	0.04632	0.17277
BPRMF($F=80, \lambda=10^{-4}$)	0.04385	0.05154	0.06262	0.19468
W-UkNN($k=500$)	<u>0.04924</u>	<u>0.05875</u>	<u>0.07224</u>	<u>0.19897</u>
W-IkNN($k=10$)	0.02165	0.03319	0.04660	0.15752
Movielens1M				
CoF($l=1, H=10, K=1040$)	0.04285 _(45.4%)	0.06952 _(41.2%)	0.10583 _(24%)	0.53959 _(0.7%)
WRMF($F=20, \lambda=1.0$)	0.02840	<u>0.04923</u>	0.07713	0.52301
BPRMF($F=160, \lambda=0.01$)	<u>0.02948</u>	0.04916	<u>0.08537</u>	<u>0.53589</u>
W-UkNN($k=80$)	0.02634	0.04577	0.07402	0.52251
W-IkNN($k=500$)	0.02446	0.04298	0.07267	0.52027
Amazon Baby				
CoF($l=1, H=3, K=54$)	0.02461 _(25%)	0.03953 _(11%)	<u>0.05924</u> _(-1.8%)	0.17624 _(0.8%)
WRMF($F=10, \lambda=10$)	0.01523	0.03272	0.05703	<u>0.17480</u>
BPRMF($F=80, \lambda=0.1$)	0.00985	0.02222	0.03902	0.16425
W-UkNN($k=500$)	<u>0.01969</u>	<u>0.03561</u>	0.06031	0.17284
W-IkNN($k=500$)	0.01616	0.02797	0.04341	0.16643

function of H . The optimal value is 40. The score is low for $H = 2$ because the data used for user group characterization is too sparse. The score is also low for $H = 200$ because too much history is included and the data does not reflect the current interests of the groups.

In CoF, we can use the whole dataset for identifying user tastes and use only recent histories for characterizing user groups and hence the items. We consider it a major advantage and it is not shared by matrix factorization. If one uses the whole dataset in MF, the item characterization vectors would be influenced by transactions long ago that do not necessarily reflect users' current interests. If one use only recent history, valuable information is lost and the data, which is typically sparse, becomes even sparser.

The choice of H also influences *global diversity*[Adomavicius and Kwon, 2012] i.e., the total number of distinct items recommended. As can be seen in Figure 4(b), the diversity decreases with H . The reason is that, as H increases, group characteristics are influenced more and more by the most active users, and consequently the algorithm becomes more and more likely to recommend items favored by those users.

Figure 4(c) and 4(d) show the Recall@5 and Diversity@5 scores respectively on Ta-feng as a function of l . We see that the scores decrease with l . As l increases, the number of latent variables decrease and the user tastes that they represent become more and more general. These results indicate that it is better to use more specific user tastes for group characterization than to use more general ones.

Just because higher levels of the hierarchy do not yield better recommendations does not mean they are useless. They can be used to define broader categories of items and be used for category-aware recommendation. For example, if a user has watched many action movies and only a small number of comedies, then we should recommend more action movies than comedies. Obviously, only a small number of categories can be used in such a strategy. Further exploration along this line is planned for future work.

7 Conclusion

Conformative filtering (CoF) is proposed as a novel method for dealing with the lack of negative examples in implicit feedback. The key assumption is that a user with a certain taste would be interested in the items that other users with the same taste have consumed before. The items not consumed by those users are hence negative examples for that taste. Taste user groups are detected using hierarchical latent tree analysis.

The assumption behind CoF is more justifiable than those behind existing alternative methods. In addition, it has several other advantages. In comparison with the user-kNN, CoF is model-based while the latter is not. When predicting the interests of a target user, CoF uses past behaviors of other users who share at least one taste with the target user, while user-kNN relies on past behaviors of only those users who share all tastes with the target user. In comparison with matrix factorization (MF), CoF uses more interpretable latent factors. Moreover, it iden-

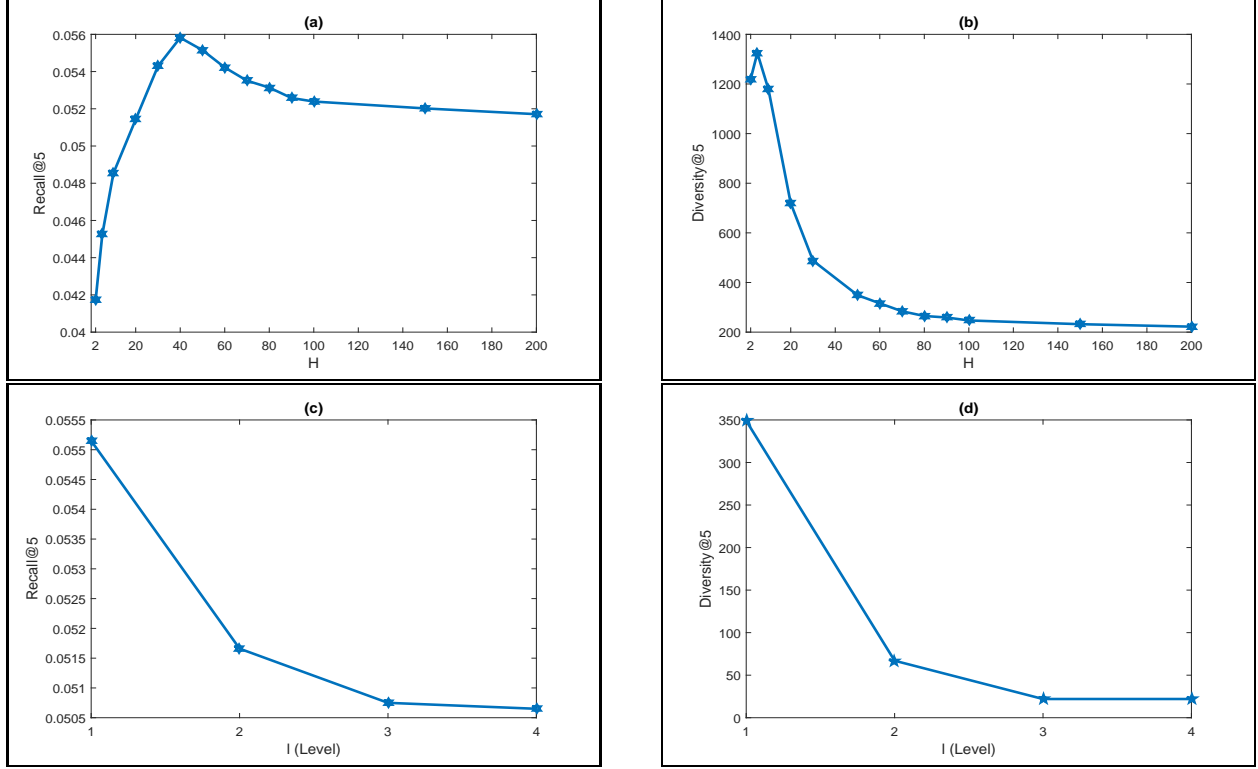


Figure 4: Impact of parameters on Recall and Diversity of CoF. (a) Impact of H on Recall for fixed l . For small H there is too little information for group characterization and for large H there is too much irrelevant information. (b) Impact of H on Diversity for fixed l . As H increases, users who consumed many items tend to increase their influence on group characterization. (c) and (d) show impact of l on recall and diversity, respectively, for fixed H . Groups become more general as l increases and highly active users tend to influence group characteristics.

tifies taste user groups using the entire consumption histories, while relying on only recent behaviors of group members to predict future interests of the group. MF does not allow such flexibility.

It has been empirically shown that CoF method substantially outperforms the alternative methods on real-world datasets from different domains. In particular, its Recall@5 and Recall@10 scores are 10% higher than those of user-kNN, and 20% higher than those of model-based methods specifically proposed to deal with implicit feedback data.

Acknowledgements

We thank Peixian Chen and Zhourong Chen for valuable discussions. Research on this article was supported by Hong Kong Research Grants Council under grants 16202515 and 16212516.

References

- [Adomavicius and Kwon, 2012] Gediminas Adomavicius and YoungOk Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. on Knowl. and Data Eng.*, 24(5):896–911, May 2012.
- [Chen *et al.*, 2012] Tao Chen, Nevin L Zhang, Tengfei Liu, Kin Man Poon, and Yi Wang. Model-based multidimensional clustering of categorical data. *Artificial Intelligence*, 176(1):2246–2269, 2012.
- [Chen *et al.*, 2016] Peixian Chen, Nevin L. Zhang, Leonard K. M. Poon, and Zhourong Chen. Progressive em for latent tree models and hierarchical topic detection. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI’16, pages 1498–1504. AAAI Press, 2016.
- [Gantner *et al.*, 2011] Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Mymedialite: A free recommender system library. In

- Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11*, pages 305–308, New York, NY, USA, 2011. ACM.
- [Goldberg *et al.*, 1992] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, December 1992.
- [Hoeffding, 1963] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.
- [Hu *et al.*, 2008] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM '08*, pages 263–272, Washington, DC, USA, 2008. IEEE Computer Society.
- [Koren and Bell, 2015] Yehuda Koren and Robert Bell. *Advances in Collaborative Filtering*, pages 77–118. Springer US, Boston, MA, 2015.
- [Liu *et al.*, 2014] Tengfei Liu, Nevin L Zhang, and Peixian Chen. Hierarchical latent tree analysis for topic detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 256–272. Springer, 2014.
- [Liu *et al.*, 2015] Teng-Fei Liu, Nevin L Zhang, Peixian Chen, April Hua Liu, Leonard KM Poon, and Yi Wang. Greedy learning of latent tree models for multidimensional clustering. *Machine learning*, 98(1-2):301–330, 2015.
- [McAuley *et al.*, 2015] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, 2015.
- [Nichols, 1997] David M. Nichols. Implicit ratings and filtering. In *Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering. Budapest: ERCIM*, volume 12, 1997.
- [Pan *et al.*, 2008] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM '08*, pages 502–511, Washington, DC, USA, 2008. IEEE Computer Society.
- [Pearl, 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.
- [Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI '09*, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.
- [Ricci *et al.*, 2010] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.
- [Ricci *et al.*, 2015] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Recommender Systems: Introduction and Challenges*, pages 1–34. Springer US, Boston, MA, 2015.
- [Wang and Blei, 2011] Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11*, pages 448–456, New York, NY, USA, 2011. ACM.
- [Wu *et al.*, 2017] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. Recurrent recommender networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM '17*, pages 495–503, New York, NY, USA, 2017. ACM.
- [Zhang and Poon, 2017] Nevin L. Zhang and Leonard K. M. Poon. Latent tree analysis. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 4891–4898, 2017.
- [Zhang, 2004] N. L. Zhang. Hierarchical latent class models for cluster analysis. *Journal of Machine Learning Research*, 5(6):697–723, 2004.

Appendix: Proof of Theorem 1

Let $x_i = 1$ when item i is picked by at least one of the m people, and 0 otherwise, and $X = \sum_{i=1}^N x_i$. Note $E(x_i) = P(x_i = 1) = 1 - (1 - \frac{1}{N})^{mn}$, so

$$E(X) = E\left(\sum_{i=1}^N x_i\right) = \sum_{i=1}^N E(x_i) = N \left(1 - \left(1 - \frac{1}{N}\right)^{mn}\right). \quad (4)$$

Since x_1, x_2, \dots, x_N are independent, by Hoeffding's inequality [Hoeffding, 1963], for $\forall t > 0$

$$P(X \leq E(X) - t) \leq \exp\left(-\frac{2t^2}{N}\right). \quad (5)$$

Choose $t = \sqrt{\frac{-N \log(1-p)}{2}}$, then $\exp(-\frac{2t^2}{N}) = 1 - p$.

If $m \geq \frac{\log\left(1 - q - \sqrt{\frac{-\log(1-p)}{2N}}\right)}{n \log(1 - 1/N)}$, we have $E(X) - t \geq qN$,

$$P(X \leq qN) \leq P(X \leq E(X) - t) \leq 1 - p \quad (6)$$

QED